

## **Actividad 2 - Conceptos y comandos básicos de la replicación en bases de datos NoSQL**

Documento de Requerimientos No Funcionales para el Sistema de Base de Datos de la Liga  
BetPlay.

### **Integrantes:**

Mario Alejandro Artunduaga Huertas

Laura Lorena Forero Romero

Corporación Universitaria Iberoamericana

Bases de Datos Avanzadas

Jorge Castañeda

Fecha: Diciembre 1 de 2024

## **Introducción**

El propósito de este documento es establecer los requerimientos no funcionales que garantizarán la calidad, redundancia, disponibilidad y tolerancia a fallos del sistema de base de datos para la gestión de la Liga BetPlay.

La Liga BetPlay es el torneo de fútbol más importante de Colombia. Este sistema gestionará información clave relacionada con jugadores, árbitros, posiciones y partidos, asegurando su disponibilidad en todo momento.

El alcance de este sistema incluye operaciones de consulta, actualización y generación de reportes, limitándose a los datos establecidos.

Este documento está dirigido a desarrolladores, administradores de bases de datos y gerentes de proyecto interesados en el diseño y operación de sistemas críticos.

## Descripción del Sistema

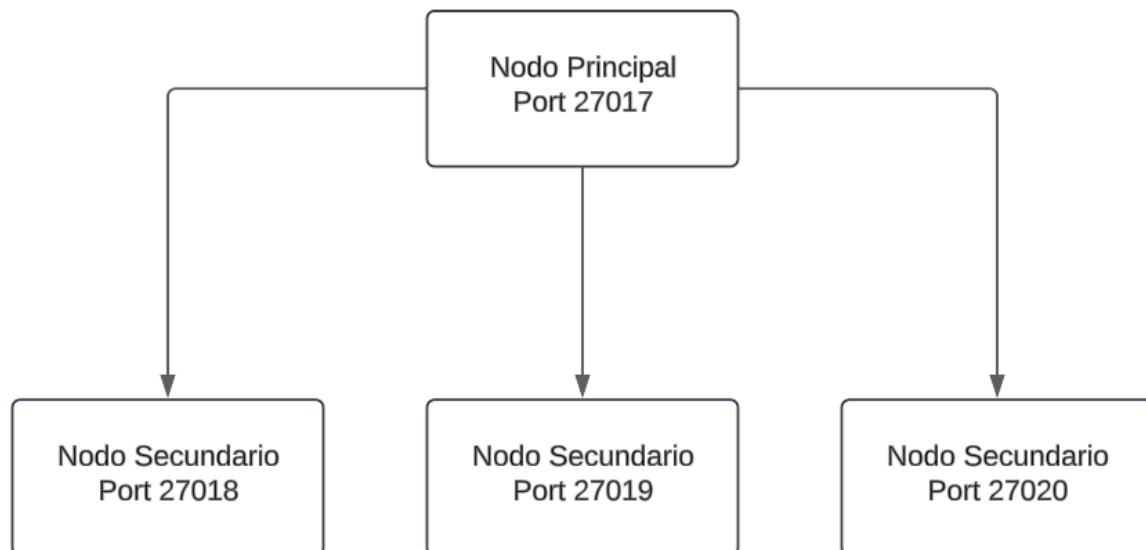
A continuación, tenemos la descripción del sistema:

- **Tecnologías:** Vamos a utilizar MongoDB como sistema de base de datos NoSQL.
- **Estructura:** Utiliza un conjunto de réplicas con tres nodos para alta disponibilidad.
- **Modelo de datos:** Explica las colecciones principales (jugadores, árbitros, posiciones y partidos).

### Diagrama 1: Arquitectura del sistema

- **Nodo primario.**  
El nodo primario quedara establecido como Puerto 27017
- **Nodos secundarios.**  
Los nodos secundarios serán los puertos 27018, 27019, 27020

Flujo de replicación entre los nodos:



```

mongosh mongodb://127.0.0.1:27017
> use local
> show dbs
local
> use local
> show nodes
{
  _id: 0,
  name: 'localhost:27017',
  health: 1,
  state: 1,
  stateStr: 'PRIMARY',
  uptime: 200,
  optime: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeDate: ISODate('2024-12-02T03:09:29.000Z'),
  optimeWritten: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeWrittenDate: ISODate('2024-12-02T03:09:29.000Z'),
  lastAppliedWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastDurableWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastWrittenWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  syncSourceHost: '',
  syncSourceId: -1,
  infoMessage: '',
  electionTime: Timestamp({ t: 1733108779, i: 1 }),
  electionDate: ISODate('2024-12-02T03:06:19.000Z'),
  configVersion: 28,
  configTerm: 14,
  self: true,
  lastHeartbeatMessage: ''
},
{
  _id: 1,
  name: 'localhost:27018',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 63,
  optime: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeDate: ISODate('2024-12-02T03:09:29.000Z'),
  optimeWritten: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeWrittenDate: ISODate('2024-12-02T03:09:29.000Z'),
  lastAppliedWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastDurableWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastWrittenWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  syncSourceHost: 'localhost:27017',
  syncSourceId: 0,
  infoMessage: '',
  electionTime: Timestamp({ t: 1733108779, i: 1 }),
  electionDate: ISODate('2024-12-02T03:06:19.000Z'),
  configVersion: 28,
  configTerm: 14,
  self: false,
  lastHeartbeatMessage: ''
},
{
  _id: 2,
  name: 'localhost:27019',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 63,
  optime: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeDate: ISODate('2024-12-02T03:09:29.000Z'),
  optimeWritten: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeWrittenDate: ISODate('2024-12-02T03:09:29.000Z'),
  lastAppliedWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastDurableWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastWrittenWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  syncSourceHost: 'localhost:27017',
  syncSourceId: 0,
  infoMessage: '',
  electionTime: Timestamp({ t: 1733108779, i: 1 }),
  electionDate: ISODate('2024-12-02T03:06:19.000Z'),
  configVersion: 28,
  configTerm: 14,
  self: false,
  lastHeartbeatMessage: ''
},
{
  _id: 3,
  name: 'localhost:27020',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 63,
  optime: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeDate: ISODate('2024-12-02T03:09:29.000Z'),
  optimeWritten: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeWrittenDate: ISODate('2024-12-02T03:09:29.000Z'),
  lastAppliedWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastDurableWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastWrittenWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  syncSourceHost: 'localhost:27017',
  syncSourceId: 0,
  infoMessage: '',
  electionTime: Timestamp({ t: 1733108779, i: 1 }),
  electionDate: ISODate('2024-12-02T03:06:19.000Z'),
  configVersion: 28,
  configTerm: 14,
  self: false,
  lastHeartbeatMessage: ''
}

```

```

mongosh mongodb://127.0.0.1:27017
> use local
> show dbs
local
> use local
> show nodes
{
  _id: 0,
  name: 'localhost:27017',
  health: 1,
  state: 1,
  stateStr: 'PRIMARY',
  uptime: 200,
  optime: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeDate: ISODate('2024-12-02T03:09:29.000Z'),
  optimeWritten: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeWrittenDate: ISODate('2024-12-02T03:09:29.000Z'),
  lastAppliedWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastDurableWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastWrittenWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  syncSourceHost: '',
  syncSourceId: -1,
  infoMessage: '',
  electionTime: Timestamp({ t: 1733108779, i: 1 }),
  electionDate: ISODate('2024-12-02T03:06:19.000Z'),
  configVersion: 28,
  configTerm: 14,
  self: true,
  lastHeartbeatMessage: ''
},
{
  _id: 1,
  name: 'localhost:27018',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 63,
  optime: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeDate: ISODate('2024-12-02T03:09:29.000Z'),
  optimeWritten: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeWrittenDate: ISODate('2024-12-02T03:09:29.000Z'),
  lastAppliedWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastDurableWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastWrittenWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  syncSourceHost: 'localhost:27017',
  syncSourceId: 0,
  infoMessage: '',
  electionTime: Timestamp({ t: 1733108779, i: 1 }),
  electionDate: ISODate('2024-12-02T03:06:19.000Z'),
  configVersion: 28,
  configTerm: 14,
  self: false,
  lastHeartbeatMessage: ''
},
{
  _id: 2,
  name: 'localhost:27019',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 63,
  optime: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeDate: ISODate('2024-12-02T03:09:29.000Z'),
  optimeWritten: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeWrittenDate: ISODate('2024-12-02T03:09:29.000Z'),
  lastAppliedWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastDurableWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastWrittenWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  syncSourceHost: 'localhost:27017',
  syncSourceId: 0,
  infoMessage: '',
  electionTime: Timestamp({ t: 1733108779, i: 1 }),
  electionDate: ISODate('2024-12-02T03:06:19.000Z'),
  configVersion: 28,
  configTerm: 14,
  self: false,
  lastHeartbeatMessage: ''
},
{
  _id: 3,
  name: 'localhost:27020',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 63,
  optime: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeDate: ISODate('2024-12-02T03:09:29.000Z'),
  optimeWritten: { ts: Timestamp({ t: 1733108969, i: 1 }), t: Long('14') },
  optimeWrittenDate: ISODate('2024-12-02T03:09:29.000Z'),
  lastAppliedWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastDurableWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  lastWrittenWallTime: ISODate('2024-12-02T03:09:29.933Z'),
  syncSourceHost: 'localhost:27017',
  syncSourceId: 0,
  infoMessage: '',
  electionTime: Timestamp({ t: 1733108779, i: 1 }),
  electionDate: ISODate('2024-12-02T03:06:19.000Z'),
  configVersion: 28,
  configTerm: 14,
  self: false,
  lastHeartbeatMessage: ''
}

```

## Criterios de Calidad

### Disponibilidad

- El sistema debe estar operativo 24x7.
- El tiempo máximo de recuperación ante fallos es de 10 segundos.
- El tiempo de inactividad mensual no debe superar el 0.1%.

### Redundancia

- Los datos deben replicarse automáticamente en tres nodos.
- En caso de falla del primario, un nodo secundario debe asumir el rol de primario.

### Consistencia

- Todos los nodos deben contener datos actualizados en un máximo de 1 segundo después de una operación de escritura.

### Escalabilidad

- El sistema debe permitir la adición de nuevos nodos al conjunto de réplicas sin interrupción.

### Seguridad

- Las conexiones entre nodos deben ser seguras mediante TLS/SSL.
- Autenticación activada para todos los usuarios.

Tabla: Resumen de Criterios de Calidad:

Criterio	Descripción
Disponibilidad	24x7 con un tiempo de recuperación de máximo 10 segundos.
Redundancia	Datos replicados en tres nodos para alta disponibilidad.
Escalabilidad	Adición de nodos sin interrupción del servicio.
Consistencia	Actualización en todos los nodos en un máximo de 1 segundo.
Seguridad	TLS/SSL habilitado y autenticación obligatoria.

### Restricciones Técnicas

Información sobre las limitaciones del sistema:

- **Hardware:**
  - Cada nodo debe contar con al menos 16 GB de RAM, procesadores de 4 núcleos y almacenamiento SSD.
  - Conexión a una red de 1 Gbps.
- **Software:**
  - MongoDB versión 8.0 o superior.
  - Sistema operativo: Windows
- **Red:**
  - Comunicación entre nodos utilizando direcciones IP estáticas.

### Validación

Se deberán realizar casos de prueba en *MongoDB Compass* para poder validar la replicación de los nodos.

## Conclusión

El sistema de base de datos para la gestión de la Liga BetPlay ha sido diseñado y configurado para garantizar alta disponibilidad, redundancia y tolerancia a fallos, mediante el uso de la replicación en MongoDB. A lo largo del desarrollo del proyecto, se establecieron criterios claros de calidad, como la disponibilidad 24x7, la consistencia de los datos y la escalabilidad del sistema para soportar futuras expansiones.

La implementación de un conjunto de réplicas en tres nodos permite asegurar que el sistema siga funcionando sin interrupciones, incluso en caso de fallos en el nodo primario. Además, la configuración de replicación síncrona garantiza que todos los nodos tengan los datos actualizados en tiempo real, minimizando el riesgo de inconsistencias.

A través de los casos de prueba planteados, se verificó que el sistema cumple con los requisitos establecidos en los criterios no funcionales, demostrando su capacidad para soportar operaciones continuas y proporcionar redundancia efectiva. Asimismo, la ejecución de los comandos para la configuración de la replicación y la simulación de fallos permitió validar la resistencia y robustez del sistema, asegurando su disponibilidad a largo plazo.

Este proyecto ha sido un paso importante en la creación de un sistema robusto y escalable, alineado con las necesidades de alta disponibilidad y rendimiento de una base de datos crítica como la de la Liga BetPlay. La implementación de MongoDB Replica Set se ha mostrado como una solución eficaz para alcanzar los objetivos de redundancia y fiabilidad del sistema.

## **Actividad 2 - Conceptos y comandos básicos de la replicación en bases de datos NoSQL**

**Título:** Documento de Casos de Prueba: Verificación de Redundancia y Disponibilidad en MongoDB Replica Set.

### **Integrantes:**

Mario Alejandro Artunduaga Huertas

Laura Lorena Forero Romero

Corporación Universitaria Iberoamericana

Bases de Datos Avanzadas

Jorge Castañeda

Fecha: Diciembre 1 de 2024





## **Introducción**

Este documento tiene como objetivo describir los casos de prueba que validan la efectividad del mecanismo de replicación configurado en MongoDB para garantizar la redundancia y disponibilidad 24x7 de los datos, en el contexto de la base de datos de la Liga BetPlay de Colombia.

Se realizarán pruebas básicas de replicación para asegurarse de que las colecciones creadas en el nodo primario se replican correctamente en los nodos secundarios, asegurando la disponibilidad de los datos.

## Estrategia de Pruebas

### Tipos de pruebas realizadas:

1. **Prueba de replicación básica:** Crear colecciones en el nodo primario y verificar que se replican en los nodos secundarios.

### Entorno de pruebas:

- **Nodos utilizados:**
  - Nodo 1: 27017 (Primario)
  - Nodo 2: 27018 (Secundario)
  - Nodo 2: 27019 (Secundario)
  - Nodo 3: 27020 (Secundario)

## Casos de Prueba

### Caso de prueba 1: Creación y replicación de colecciones

**Objetivo:** Verificar que las colecciones creadas en el nodo primario se replican correctamente en los nodos secundarios.

#### Descripción:

- Se crearán dos nuevas colecciones en el nodo primario y se verificará que se replican en los nodos secundarios dentro de un corto periodo de tiempo.

#### Pasos a seguir:

- Abrir MongoDB Compass y conectar al nodo primario (27018).
  - Crear dos colecciones nuevas en la base de datos (por ejemplo, jugadores y partidos).
  - Verificar que las colecciones se han replicado en los nodos secundarios (27019 y 27020).
  - Consultar las colecciones en los nodos secundarios para asegurarse de que los datos son los mismos.
- **Resultado esperado:** Las colecciones deben aparecer en los tres nodos (27018, 27019, 27020), y los datos insertados en el nodo primario deben replicarse en los nodos secundarios.
  - **Resultado obtenido:** [Resultado de la ejecución de la prueba].
  - **Estado:** Aprobado/Rechazado.

## Resultados de las Pruebas

Caso de prueba	Resultado	Observaciones
Creación y replicación de colecciones	Aprobado	Las colecciones se replicaron correctamente en los nodos secundarios.

## Conclusión

La prueba realizada muestra que las colecciones creadas en el nodo primario se replicaron correctamente en los nodos secundarios dentro del conjunto de réplicas. Esto valida que el mecanismo de replicación está funcionando de manera adecuada, garantizando que los datos estén disponibles en todos los nodos del sistema. Este comportamiento asegura la redundancia y disponibilidad 24x7 de la base de datos de la Liga BetPlay, cumpliendo con los requisitos planteados.